CMPSC 465 — Data Structures & Algorithms
Spring 2024 — Mehrdad Mahdavi and David Koslicki

Worksheet 10

Wednesday, Mar 27, 2024

1. **Proof Technique Review.** Please review the following common proof techniques and highlight how they are used in this week's problems:

   (a) Direct Proof (Problem 3)

   (b) Proof by Contradiction (Problem 2)

   (c) Proof by Contrapositive (No problem, just know that $a \to b$ is equivalent to $\neg b \to \neg a$)

   (d) Proof by Induction (Problem 6)

2. **Prefix-Free Encoding.** Given a finite alphabet Γ, a prefix-free encoding assigns each symbol in Γ a binary codeword such that no codeword is a prefix of another codeword. A prefix-free encoding is minimal if it is not possible to arrive at another prefix-free encoding by reducing any of the codewords. For example, the encoding $0, 101$ is not minimal because the codeword 101 can be reduced to 1, and the encoding would still be prefix-free.

   Show that any minimal prefix-free encoding can be represented by a full binary tree in which each leaf corresponds to a unique element of Γ, and each codeword can be generated by tracing the path from the root to that symbol's leaf.

3. **Service Scheduling.** A server has n customers waiting to be served. Customer i requires t_i minutes to be served. If, for example, the customers were served in the order $t_1, t_2, t_3, \ldots$, then the ith customer would wait for $t_1 + t_2 + \cdots + t_i$ minutes.

   We want to minimize the total waiting time

$$T = \sum_{i=1}^{n} (\text{time spent waiting by customer } i)$$

   Given the list of t_i, give an efficient algorithm for computing the optimal order in which to process the customers.

4. **Specified Leaves MST.** Given a connected, undirected graph $G = (V, E)$ with edge weights w_e and a subset of vertices $U \subset V$, give an $O(|V| + |E| \log |V|)$ algorithm to determine the lightest spanning tree T such that every $u \in U$ is a leaf node in T.

5. **Longest Encoding.** Under a Huffman encoding of n ($n \geq 2$) symbols with frequencies $f_1, f_2, \ldots, f_n$ (assuming they are in non-increasing order), what is the longest length a codeword could possibly have? In what kind of case(s) (specifically, in terms of n and $f_1, f_2, \ldots, f_n$) can this happen? Show how you derive those case(s).

6. Fibonacci Frequencies.

(a) What is an optimal Huffman code for the following set of frequencies, based on the first 8 Fibonacci numbers?

$$a:1 \qquad b:1 \qquad c:2 \qquad d:3 \qquad e:5 \qquad f:8 \qquad g:13 \qquad h:21$$

(b) Generalize your answer to give the optimal encoding when the frequencies are the first n Fibonacci numbers. Prove that your encoding is optimal by showing that it is produced by the Huffman encoding algorithm. (*Hint*: Consider relating Fibonacci number $k+1$ to the sum of the first $k-1$ Fibonacci numbers).